DTD Level authorization in XML documents with usage control

Lili Sun and Yan Li
Department of Mathematics and Computing
University of Southern Queensland, Toowoomba QLD 4350 Australia

Summary
In recent years an increasing amount of semi-structured data has become important to humans and programs. XML promoted by the World Wide Web Consortium (W3C) is rapidly emerging as the new standard language for semi-structured data representation and exchange on the Internet. XML documents may contain private information that cannot be shared by all user communities. So securing XML data is becoming increasingly important and several approaches have been designed to protect information in a website. However, these approaches typically are used at file system level, rather than for the data in XML documents. Usage control has been considered as the next generation access control model with distinguishing properties of decision continuity. Usage control enables finer-grained control over usage of digital objects than that of traditional access control policies and models.

In this paper, we present a usage control model to protect information distributed on the web, which allows the access restrictions directly at DTD-level and XML document-level. Finally, comparisons with related works are analysed.

Key words:
XML documents, the Document Type Definition(DTD), Usage control, Authorization

1. Introduction

Over the past several years, there has been a tremendous surge of interest in XML as a universal, queryable representation for data. XML web service is a platform-independent web application that accepts requests from different systems on the Internet. XML is a fundamental component in many XML web services and it is used to store and exchange data in the Internet environment that may include private messages of customers. It overcomes the complexity of SGML and the user can define document structures, removing the limit of the fixed tags in HTML. The following example displays customer's information in a XML document.

Manuscript received November 5, 2006.
Manuscript revised November 25, 2006.

```
<xml version= "1.0" encoding= "UTF8"?>
   <customerInfo xmlns=
      "http://www.bookstore.com/BooksInfo">
     <bookstore city= "Toowoomba">
       <books>
        <available>
        <categorize> textbook </categorize>
        <price >$22.00 </price >
        <exercisebook>
           <description >
           <English comprehensive>
           </description >
           <price > $18.00 </price >
        </exercisebook >
        </available >
        </sold >
        <categorize >
            magazine
        </categorize>
        <price > $30.00 </price>
        <buyer>
           <name > Tony </name >
           <address> Jilan street, 5
           </address>
           <city> Toowoomba </city>
        </buyer>
        </sold>
       </books>
     </bookstore>
   </customerInfo>
```

Table 1: XML Document Example

XML documents not only show the content of data but also the constraints and relationships between data. In Table 1, the element *customerInfo* includes *bookstore*, and *books* sub-elements. The sub-element *available* is a simple type while sub-elements *model* and *price* are combined with their own sub-elements. Since an XML document can express complex relationship between data, it may be generated from various resources with varying security requirements. In some situations a user may like to access the particular parts of an XML document. In the above example, for the textbook objects everyone can read all information. However, some users' access to information such as *sold* and *buyer* will be restricted. This is because when an internal or external user accesses this document, his/her access permission has to be limited according to security policies in all databases. This example shows that secure XML documents form a significant topic for research.

In general, we identify two levels, instance and DTD (the Document Type Definition) at which authorizations on XML documents can be defined. XML documents and DTDs naturally support two levels of authorization. 1). Low-level authorizations, associated with XML documents, providing full control of authorizations on a document by document basis; 2). High-level authorizations, associated with XML DTDs, providing structure and element declarations of access permissions. Different requirements may have call for the support of access restrictions at the level of each specific document [2]. In the access control model the central authority uses XML DTDs to specify the format of information to be changed.

Several approaches have been designed for the security of XML documents [5, 6, 14]. But all these approaches have some limitations. Encryption and decryption skills [14] focus on the protection at the file level not on a systematic level. They are used in protection of communications between servers and clients rather than dissemination from clients. Traditional access control models primarily consider static authorization decisions based on the subjects' permissions on target objects, they have used on the control of access to server-side objects. From these models they present a similar approach in the work: a security administrator defines a set of policies at document level or DTD level. Through access control, the system can restrict unauthorizatied users access to the resources in the system and guarantee the confidentiality and integrity of the resources. On the other hand, traditional authorization decisions are generated at request time but do not consider ongoing controls for long access or for revocation. Recently proposed usage control [16] is a new access control model extending traditional access control models in multiple aspects. The main different properties of usage control with traditional access control models are continuity of access decision and mutability of subject attributes and object attributes.

In this paper, we propose authorization models which adopt usage control to manage access both at the instance-level and at the schema-level. Following traditional access control given an access request, an algorithm computes a view of the target XML document based on the user's requirements right. It has analysed authorization decisions on a subject's access to target resources before access. In usage access control authorization decisions are not only checked and made before access, but also are repeatly checked during the access period. It may revoke access permission according to the changes of the subject or object attributes. Meanwhile obligations and conditions become decision factors for the management of XML documents.

The remainder of this paper is organized as follows: Section 2 illustrates the background of XML and DTD level authorizations. Section 3 presents the usage control model. Three decision factors: *Authorization*, *Obligations*, *Conditions* and *Continuity* properties *pre* and *ongoing* are introduced in this section. Section 4 shows our proposed authorization models for usage control. It includes pre-*Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions* and *ongoing-Conditions.* Section 5 concludes the paper and outlines our future work.

2. Related technologies

2.1 DTD and XML documents

XML [3] is a markup language for describing semi-structured information. An XML document consists of elements, attributes and text nodes, each delimited by a pair of start and end tags (e.g. <price> and </price>) or by an empty tag. The content of each element is a sequence of elements or text nodes. An element has a set of attributes, each of which has a name and a value. XML document can be classified into two categories: well-formed and valid. A document is said to be well-formed if it follows the grammar rules of XML, such as there is exactly one element that completely contains all other elements, elements may nest but not overlap, etc. A well-formed document is valid only if it contains a proper DTD in the source and if the document obeys the constraints of that declaration. Validation requires an XML instance to contain specified elements and attributes, following specified datatypes and relationships.

Document Type Definition (DTD) and XML Schema are two main validation specification mechanisms. A DTD is a file which contains a formal definition of a particular type of XML document. A DTD consists of two parts: the element declarations and the attributes declarations. Elements are the most important components of an XML document. Element declarations in the DTD specify the names of elements and their contents. They also describe sub-elements and their cardinality. Attributes represent properties of elements. Attribute declarations specify the attributes of each element, including their name, type, etc. Attributes can be marked as *required*, *implied*, or *fixed*. Attributes with *require* must have an explicit value for each occurrence of the associated elements. Attributes with *implied* are optional. Attributes with *fixed* have a fixed value. Entities are used to include texts and binary data. Notations specify how to manage entities and binary data. Entities and notations are not considered in this paper since both of them are used to describe the physical structure of an XML document.

However, the XML documents corresponding to a DTD must obey a structure defined by that DTD. Each DTD is a schema and XML documents corresponding to that DTD are instances of that schema. But the DTD structure is not restricted. For instance, two different XML documents with the same schema may widely differ in the elements. As a well-formed XML document is in nested structure, there are some languages (e.g., XHTML and XML DOM) which are applied to locate elements with patterns. An XML document can be generated from various resources to fit applications with different structures. These main aspects about XML are discussed in [8]. The example below in Table 2 displays an XML DTD for a corresponding valid XML document in Table 1.

```
<?xml version= "1.0"  encoding= "UTF-8"? >
    <xs:annotation>
        <xs:documentation>
            Customer Information Instance
        </xs:documentation>
    </xs:annotation>
    <xs:ELEMENT bookstore (book+)>
    <xs:ELEMENT books (available*, sold*)>
    <xs:ELEMENT available (catalog, price,
                          exercisebook*, buyer)>
    <xs:ELEMENT sold (categorize, price, buyer)>
    <xs:ELEMENT exercisebook (PCDATA)>
    <xs:ELEMENT buyer (name, address, city, discount?)>
    <xs:ELEMENT categorize (PCDATA)>
    <xs:ELEMENT price (PCDATA)>
    <xs:ELEMENT description (PCDATA)>
    <xs:ELEMENT name (PCDATA)>
    <xs:ELEMENT address (PCDATA)>
    <xs:ELEMENT city (PCDATA)>
    <xs:ELEMENT discount (PCDATA)>
    <xs:ATTLIST bookstore city CDATA #REQUIRED>
```

Table 2: XML DTD Example

2.2 Usage control

The traditional access control method normally deals only with authorization decisions on users' access to target resources. The usage control is a generalization of access control. It enriches and refines the access control discipline in its definition. There are eight core components in the usage control model: subjects, subject attributes, objects, object attributes, rights, authorizations, obligations, and conditions [16, 20] (see Figure1). In the usage control model, subjects and objects are familiar concepts with traditional access control. A right represents access of a subject to an object, such as read or write. The existence of the right is determined when the access is attempted by the subject. Obligations and conditions are new concepts that can resolve certain shortcomings that have been in traditional access controls. Obligations are requirements that have to be fulfilled by obligation subjects for allowing access. Conditions are subject and object independent environmental or system requirements that have to be satisfied for access.

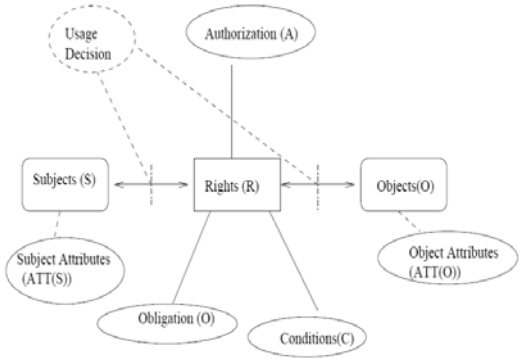

Fig.1 Components of Usage Control Model

The usage decision functions indicated in Figure1 make this determination based on subject attributes, object attributes, authorizations, obligations and conditions at the time of usage requests.

Subjects, objects, and rights can be divided into detailed components with different perspectives. A subject can be a user, a group, a role, or a process. In the usage control model, the subjects can be consumer subjects (CS), provider subject (PS), and identifier subjects (IS). Objects are entities that subjects hold rights on, whereby the subjects can access or use objects. Rights are privileges that subjects can hold on objects. The authorization of rights requires associations with subjects and objects.

Subject and object attributes can be used during the access decision process. Subject attributes are identities, group names, roles, memberships, security clearance and so on. An online shopping buyer, an university student in management system can both be subjects. Object attributes associated with objects are security labels, ownerships, classes, access control lists and so on. For instance, in an on-line shopping store, a price could be an object attribute, the electron blood pressure is priced at $120 and with delivery is required at $135.

Authorizations, obligations and conditions are decision factors used to check and determine whether a subject should be allowed to access an object. Authorizations are based on subject and object attributes and the specific right. Authorizations can be either pre-authorization (preA) or ongoing-authorization(onA). Pre-authorization is performed before authorization is required to the access.

In general, the authorization of most traditional access controls are assumed to be done before access is allowed (pre). But ongoing authorization may be performed during the access, such as when a book stocking list in a bookstore is periodically checked while the access is in progress. An access is immediately revoked if the relevant item's number becomes 0 when it appears on the list. However, it is quite reasonable to extend this for continuous enforcement by evaluating usage requirements throughout usages (ongoing). Authorizations may require updates on subject and object attributes. These updates can be either `pre', `ongoing', or `post' and are called "continuity properties". Figure 2 shows the continuity properties in usage control model.

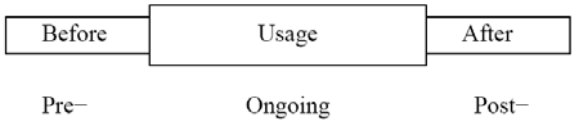

Fig.2 Continuity Properties of Usage Control

Obligations are requirements that a subject must perform before (pre) or during (ongoing) accesses. An example of a pre-obligation is the requirement that a user must provide some contact and personal information before accessing IEEE digital library. The requirement that a user has to keep certain advertising windows open while he is accessing some service, is an example of an ongoing obligation. Subject and object attributes can be used to decide what kind of obligations are required for access approval.

Conditions are decision factors that depend on environmental and system-oriented requirements. Subject and object attributes can be used to select which condition requirements have to be used for a request. For example, IEEE member can access full papers in the IEEE digital library. They can also include the security status of the system, such as low level, normal, high alert.

3. Authorization models

In this section we consider authorization models for the DTD and XML documents adopting usage control. Based on the involvement of three decision factors: authorizations, obligations, and conditions, we develop models for usage control which consider on enforcement. We assume that a usage request exists on an XML target object. Decision-making can be done either before (pre) or during (ongoing) exercise of the requested right. Decision-making after the usage has no influence on the decision of current usage. Based on the requirements we have six possible cases as a model for usage control: pre Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions and ongoing-Conditions. Depending on the access requirements on the DTD and XML documents in the real world, it is possible to utilize more than one case. In this paper, we consider only the" cases consisting of Authorizations, Obligations or Conditions alone with pre or ongoing decisions. Meanwhile we focus on developing the usage control models for the DTD and XML documents.

A. Usage control for pre-Authorization Model UCM_preA:

In a pre-Authorization usage control model, the decision process is performed before access is allowed. Consider the XML DTD and XML documents in Figure 1 and Figure 2. The following illustration of usage decision that can be expressed on the documents DTD level and instance level are made in pre-authorizations.

The UCM_preA model consists of the following components: S, XDTD, XD, R, R1, ATT(S), ATT(XDTD), ATT(XD) and usage decision Boolean functions preA, preA_1 on XDTD, XD, respectively, where S, XDTD, XD, R, R1 represent Subject, XML DTD, XML document and Rights required on XML DTD level and XML document respectively. ATT(S), ATT(XDTD), ATT(XD) represent attributes of subjects, XML DTD and XML document respectively. preA and preA_1 are predicates about authorization functions.
For example, consider the XML DTD and the XML document in Figure 1 and Figure 2:
preA in DTD level is applicable to all bookstores.
D1: Information about the available books is publicly accesible for every reader.
D2: Information about the price sold books is only accesssible by administrative staff.
D3: The original price of books is not publicly accessible.
preA_1 in instance level is applicable to a bookstore in Toowoomba.
L1: The price of sold books is only accessible by members of financial staff.
L2: Tony can access information about books but those books that have been discounted and sold in other cities.

1. allowed (s, xdtd, r) $\Rightarrow$ preA(ATT(s), ATT(xdtd), r),
   where A $\Rightarrow$ B means B is a necessary condition for A.
   In this example this predicate indicates that if subject S is allowed to access XML DTD level xdtd with right r then the indicated condition preA must be match D1, D2,
   D3.

2. allowed (s, xd, r1) $\Rightarrow$

preA_1(ATT(s), ATT(xd), r1).
In this example the allowed (s, xdl, r1) predicate indicates that if subject s is allowed to access XML document xd with right r1 then the decision function preA_1 must be matched with L1 and L2.

The UCM_preA model provides an authorization method on whether a subject can access the XML DTD level and Instance level document. The allowed(s, xdtd, r) predicate shows that subject s can access information in the XML DTD level document. The allowed(s, xd, r1) predicate shows that subject S can access information in XML documents. At this process, private information in XML DTD and corresponding XML documents are restricted.

B. Usage control for ongoing Authorizations Model UCM_onA:

A usage control model for ongoing-Authorizations model is used to check ongoing authorizations during access processes. In this model, usage requests are allowed without any 'pre' decision making.

The UCM_onA model has the following components:
S, XDTD, XD, R, R1, ATT(S), ATT(XDTD), ATT(XD) as before, and ongoing usage decision functions onA on XDTD (XML DTD level) and onA_1 on XD (XML document).
onA and onA_1 are used to check whether S can continue to access or not.

1. allowed (s, xdtd, r) $\Rightarrow$ true,
   This is a prerequisite for ongoing authorization on xdtd.

2. allowed (s, xd, r1) $\Rightarrow$ true,
   This is a prerequisite for ongoing authorization on xd.

3. stopped (s, xdtd, r) $\Leftarrow$ ┐onA(ATT(s), ATT(xdtd), r),
   The access of subject s to xdtd is terminated if the ongoing authorization onA is failed.

4. stopped (s, xd, r1) $\Leftarrow$
   ┐ onA_1(ATT(s), ATT(xd), r1).
   The access of subject s to xd is terminated if the ongoing
   authorization onA_1 is failed.

In this model usage decision Boolean function are *onA, onA_1* instead of preA, preA1. During this process the requested access is always allowed as there is no pre-authorization all the time. allowed (s, xdtd, r) and allowed (s, xd, r1) are required to be *true*, otherwise ongoing authorization should not be initiated. Ongoing authorizations are active throughout the usage of the requested right, and some requirements are repeatedly checked for continued access. These checks are performed periodically based on time or event. In the process when attributes are changed and requirements are no longer satisfied, stopped procedures are performed. Stopped (s, xdtd, r) and stopped (s, xd, r1) indicate that rights r and r1 of subject s on object XML DTD and XML document are revoked and the ongoing access terminated. For example, a limited number of simultaneous usage, suppose only two administration staff can access information about the price sold books in an object XML DTD level simultaneously. If a third administration staff requests access and pass the pre-authorization, the staff with the earlier time access is terminated. While this is a case of ongoing authorizations, it is important that the certificate should be evaluated in a *pre* decision.

C. Usage control for pre-Obligations Model UCM_preB:

UCM_preB introduces pre-obligations that have to be fulfilled before access is permitted. It will return true or false for usage decision depending on whether obligation actions have been fulfilled or not. This model consists of two steps: the first is to select required obligation elements for the requested usage, and then to evaluate whether the selected obligation elements have been fulfilled or not. Examples of pre-obligations are requiring a reader to register by filling forms before accessing online reading, and requiring a reader to click the ACCEPT box on a license agreement to read some books online. The pre-obligation action may perform on a different object (e.g., register, license agreement) that the reader is trying to access (e.g., e-book). It means that the pre-obligation action may be done by some other subject. Hence obligation subjects, objects, and actions are added in the following UCM_ preB model.

The following model is described for XML DTD level and XML documents. It can be used for restricted information in XML DTD or in the XML documents.
The UCM_preB model has the following components:
S, XDTD, XD, R, ATT(S), ATT(XDTD) and ATT(XD) are as before, OBS, OBO and OB represent obligation subjects, objects, and actions, respectively; decision function *preObfilled* : OBS × OBO × OB $\rightarrow$ {true, false}, As mentioned above, subject S and access object XDTD, XD may be different from OBS and OBO.
The function *preObfilled*(s, xdtd, r) and *preObfilled*(s, xd, r) are used to check if obligations are obeyed or not before the subject(s) accesses the object(xdtd, or xd).

1. allowed(s, xdtd, r) $\Rightarrow$ preObfilled(s, xdtd, r).
   The preObfilled(s, xdtd, r) function must be true if subject(s) is allowed to access xdtd with right r.

2. allowed(s, xd, r) $\Rightarrow$ preObfilled(s, xd, r) .
The preObfilled(s, xd, r) function must be true if s is allowed to access xd with right r.

This model indicates that obligations have to be fulfilled before s can access xdtd or xd. Note that each obligation has to be true if there are more than two obligations.

D. Usage control for ongoing-Obligations Model UCM_onB:

Similar to pre-Obligations, Obligations are required to fulfill in UCM_onB models while rights are exercised. Ongoing-obligations may have to be fulfilled periodically or continuously. For example, when a reader accesses an e-book through the Internet within every 15 web pages, the reader may have to open an advertisement window. Alternatively, the reader may leave an advertisement window active all the time with inconvenience. The model concerns obligations that have to be fulfilled.

The UCM_onB model has the following components:
S, XDTD, XD, R, ATT(S), ATT(XDTD) and ATT(XD) as before, OBS, OBO, and OB represent obligation subjects, objects, and actions, respectively; an ongoing decision function onObfilled : OBS $\times$ OBO $\times$OB $\rightarrow$ {true, false}. The ongoing function preObfilled(s, xdtd, r) and preObfilled(s, xd, r) are used to check if obligations are continually obeyed or not during subject(s) access object(xdtd or xd).

1. allowed(s, xdtd, r) $\Rightarrow$ true ,
A prerequisite for UCM_onB. It means that s is accessing XML DTD.

2. allowed(s, xd, r) $\Rightarrow$ true,
A prerequisite for UCM_onB. It means that s is accessing XML document.

3. stopped (s, xdtd, r) $\Leftarrow \neg$ onObfilled(s, xdtd, r).

4. stopped (s, xd, r) $\Leftarrow \neg$ onObfilled(s, xd, r).

Where stopped (s, xdtd, r) indicates that the access of s on xdtd with r is revoked if the ongoing obligations fail. Alternatively, stopped (s, xd, r) indicates that the access of s on xd with r is revoked if the ongoing obligations fail.

E. Usage control for pre-Conditions Model UCM_preC:

As described earlier, conditions define that certain restrictions have to be satisfied for usages. Conditions are not directly related to subjects and objects since they define environmental and system restrictions. By using conditions in usage decision processes, it can provide finer-grained controls on usage. We focus on this model on XML DTD and XML documents. The pre-conditions model has to be used before requested rights are exercised. For example, suppose there are some requirements to restrict times for reading papers, such as papers can be read 5 times, print 3 times. You then should check them before a usage allowed.

The UCM_preC model has the following components:
S, XDTD, XD, R, ATT(S), ATT(XDTD) and ATT(XD) as before, preCON (a set of pre-conditions), verify conditions function preConSatisfied: preCON $\rightarrow$ {true, false}, The function preConSatisfied is used to check whether the pre-conditions are satisfied or not.

1.preC(s, xdtd, r) =
$\wedge_{\text{preCon\_i} \in \text{preCON}}$ preConSatisfied(preCon_i),or

2. preC(s, xd, r) =
$\wedge_{\text{preCon_i} \in \text{preCON}}$preConSatisfied(preCon_i).
All pre-conditions have to be checked if there are more than two conditions.

3. allowed(s, xdtd, r) $\Rightarrow$ preC(s, xdtd, r), or

4. allowed(s, xd, r) $\Rightarrow$ preC(s, xd, r).

where allowed(s, xdtd, r) and allowed(s, xd, r) express that all conditions have to be satisfied before access is approved.

F. Usage control for ongoing-Conditions Model UCM_onC:

UCM_onC model requires conditions to be satisfied while rights are in active use. If violating any of the restrictions, the allowed right is revoked and the exercised is stopped. For example, realOne player does not work when Windows XP system works on safety module.

The UCM_onC model has the following components:
S, XDTD, XD, R, ATT(S), ATT(XDTD) and ATT(XD) as before, onCON (a set of ongoing conditions), verify ongoing conditions elements.
onConSatisfied: onCON $\rightarrow$ {true, false}.
The function onConSatisfied is used to check whether ongoing conditions are satisfied or not.

1. onC(s, xdtd, r) =
$\wedge_{\text{onCon\_i} \in \text{onCONon}}$ ConSatisfied(onCon_i), or

2. onC(s, xd, r) =
$\wedge$ onCon_i $\in$ onCON onConSatisfied(onCon_i),
All ongoing conditions are required to be checked.

3. allowed(s, xdtd, r) $\Rightarrow$ true, or

4. allowed(s, xd, r) $\Rightarrow$ true,
A prerequisite for UCM_onC.

5. stopped (s, xdtd, r) $\Leftarrow \neg$ onC(s, xdtd, r).

6. stopped (s, xd, r) $\Leftarrow \neg$ onC(s, xd, r).

In practice, the above six models may need to be combined for an access control. We obtain an authorization method for XML DTD and XML documents and their elements by checking users' (subjects') authorizations, obligations and conditions with continuity properties.

4. Conclusions and future work

In this paper we introduce DTD, XML, usage control and discuss access models for XML DTD and XML documents by using usage control. Usage control models provide an approach for the next generation of access control. We analyse not only decision factors in usage control, such as authorizations, obligations and conditions, but also the continuity properties. This paper also illustrates six different kinds of models built for XML DTD and XML documents. In addition, the work in this paper has significantly extended previous work, such as the ongoing continuity for authorizations, obligations and conditions in usage control for XML DTD and XML documents. The methods presented in this paper can be used to control XML documents in a dynamic environment. It also begins a new application with usage control.

Obviously, there is an increasing realization that traditional access control is not adequate for modern application needs. This paper represents only a first step for DTD level authorization in XML documents with usage control, and much work is still to be done before these models can be used in practice.

References

[1] Bertion E. and Ferrari E. Secure and selective dis-semination of xml documents. *ACM trans. Inf. Syst. Secur.*, 5(3):290-331, 2002.

[2] Bertion E., Castano S., Ferrari E. and Mesiti E. Controlled access and dissemination of xml documents. In *Processings of the second international workshop on Web information and data management*, pages 22-27. ACM Press, 1999.

[3] Bray T., Paoli J., Sperberg M and Maler E. *Extensible Markup Language (XML) 1.1 (Second Edition)*. World Wide Web Consortium (W3C), Cambridge, MA, USA, 2000.

[4] Chen F. and Sandhu R. Constraints for role-based access control. In *Proc. First ACM Workshop on Role-based Access Control*, pages 39-46, 1995.

[5] Damiani E., Capitani S. and Samarati P. Towards securing xml web services. In *Proc .of the 2002 ACM Workshop on XML Security,* Washington, DC, USA, November 2002.

[6] Damiani E., Paraboschi S. and Samarati P. A fine-grained access control system for aml documents. *ACM Trans. Inf. Syst. Secur.,* 5(2):169-202, 2002.

[7] Damiani E., Vimercati S., Paraboschi S. and Samarati P. Design and implementation of an access processor for xml documents. *In Processings of the 9th international WWW conference*. Amsterdam, 2000.

[8] Damiani E., Vimercati S., Paraboschi S. and Samarati P. Securing xml documents. *In Lecture Notes in Computer Science*, volume 1777, pages 121-135. Springer, 2000.

[9] Ford W. and Baum M. S. Secure electronic commerce: *Building the Infrastructure for Digital Signatures & Encryption*. Prentice Hall PTR, 1997.

[10]Gabillon A. An authorization model for xml databases. In *Proceedings of the 11th ACM conference on Computer Security*, 2004.

[11] Gabillon A. and Bruno E. Regulating access to xml database. In *Proc. 15th Ann. IFIP WG 11.3 Working Conf. Database Security*, July 2001.

[12] Igor T., Zachary G.I., Alon Y.H and Daniel S.W. Updating xml. 2001.

[13] Kudo M. and Hada S. Xml access control.

[14] Kudo M. and Hada S. Xml document security based on provisional authorization. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 87-96. ACM Press, 2000.

[15] Lim C. H., Park S. and son S.H. Acess control of xml documents considering update operations. *ACM Workshop on XML Security*, 2003.

[16] Park J. and Sandhu R. Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, page 57-64. ACM Press, 2002.

[17] Sandhu R., Conyne E., Feinstein H. and Youman C. Role-based access control models. In *IEEE Computer*, number 2, pages 38-47, February 1996.

[18] Sun L. and Li Y. Xml undeniable signature. In *Proceedings of International Conference Computational Intelligence for Modelling, Control and Automation*, IEEE, 2005.

[19] Wang Y. and Tan K. A scalable xml access control system. In *Proceedings of the 10th international WWW conference*. Poster, 2001.

[20] Zhang X., Park J. and Parisi-Presicce F. A logical specification for usage control. In *SACMAT'4*. ACM Press, 2004.

[21] Zhang X., Park J. and Sandhu R. Schema based xml security: Rbac approach. In *Proceedings of the IFIP WG*. ACM Press, 2003.